

35. (New) A method of providing a lexical cache, comprising the computer-implemented steps of:

B2 allocating a plurality of lexical containers each configured to a respective maximum number of entries based on a key length; and searching for one of the entries associated with a string within one of the plurality of lexical containers corresponding to a key generated based on the string, wherein at least one of the lexical containers is configured to hold a different number of the entries than at least another one of the lexical containers.

36. (New) The method of claim 35, wherein the maximum number of entries in the allocating step is based on a logarithmic function of the key length.

#### REMARKS

By this amendment, claims 1-36 are pending, in which claims 35 and 36 are newly presented, and claims 1, 5, 15-16, 20, and 30-32 are amended. Care was exercised to avoid the introduction of new matter.

The Office Action mailed May 31, 2002 rejected claims 1-34 as obvious under 35 U.S.C. § 103 based on *Li* (US 5,774,588) in view of *Cohen* (US 6,169,969). This rejection is respectfully traversed because neither *Li* nor *Cohen* teach or otherwise suggest the limitations of the claims, as amended.

For example, independent claims 1, 16, 31-32, as amended, each recites that the lexical containers are “associated with respective key lengths and configured to hold **respective maximum numbers of entries based on the respective key lengths**” and that “at least one of the lexical containers is configured to hold a **different number of entries** than at least another

one of the lexical containers.” Both independent claims 15 and 30 now recite that “each of **said sequences of slots** corresponding to a respective hash value and a number of the slots being **based on a respective key length**, wherein at least one of the hash tables is configured to hold a **different number of slots** than at least another one of the hash tables.” //

Provision, in a lexical cache, of lexical containers or hash tables configured to hold a maximum number of entries or slots based on respective key lengths advantageously enables each lexical container or hash table to be individually tuned so that “the lexical cache by its structure will tend to contain more frequently accessed words.” (specification, p. 3). For example, as recited in dependent claims 33-34, a lexical container associated with a shorter key can be configured to hold more entries than a lexical container associated with a longer key. The references of *Li* and *Cohen*, either individually or in combination, fail to satisfy these features and limitations.

As explained in the previous response dated May 6, 2002, *Li et al.* is directed to a method for comparing strings with entries of a lexicon with a fixed-length key, and does not disclose a method in which a lexical container or hash table from among a plurality of lexicon based on “a length of the key” as recited in the claims. The signature vector **25** key is fixed length at 85 bits, and there thus is no need or motivation in *Li et al.* to use that key’s constant length to identify a particular lexical container or hash table.

In the rejection of claims 1, 16, and 31-32, the Office Action acknowledges that the primary reference *Li* fails to disclose the claimed feature of a length of the key. Consequently, the Office Action, on page 4, applies the newly cited *Cohen*, asserting that *Cohen* “strongly teaches step of calculating m delaying additively complementary to the m n-gram selection positions respectively for the distinct length of the keywords in the dictionary,” referring to col. 4, lines 50-53.

However, *Cohen* discloses, per col. 4, lines 13-63, a dictionary string matching method for locating all matches of a keyword dictionary in a sample byte stream, in which *m* binary-valued hash tables are built and each entry, called a presence value, is set to a logical false. There is no disclosure that the hash tables is configured to hold a different number of key values than at least another one of the hash tables, in the manner claimed. Further, as seen in FIG. 1 of the reference, the hash tables 24 are commonly labeled, indicating that no distinction is being made as to their characteristics – that is, these hash tables 24 are not different from one another.

Dependent claims 2-14, 17-29, and 33-34 are allowable for at least the reasons put forth for the allowability of their corresponding independent claims and are separately patentable on their own merits.

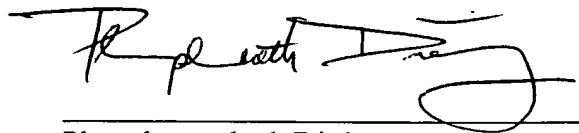
With respect to new claims 35-36, independent claim 35 is directed to a method of providing a lexical cache, and recites “allocating a plurality of lexical containers each configured to a respective maximum number of entries based on a length of a key” and that “at least one of the lexical containers is configured to hold a different number of the entries than at least another one of the lexical containers.” As acknowledged by the Office Action, the reference of *Li* is silent regarding the feature of “a length of a key.” Further, the *Cohen* reference cannot satisfy these features and limitations, in part, for the reasons argued above with respect to claims 1, 16, 31-32. Furthermore, neither reference discloses that “the maximum number of entries in the allocating step is based on a **logarithmic function** of the key length” as recited in claim 36.

Therefore, the present application, as amended, overcomes the rejections of record and is in condition for allowance. Favorable consideration is respectfully requested. If any unresolved issues remain, it is respectfully requested that the Examiner telephone the undersigned attorney at 703-425-8516 so that such issues may be resolved as expeditiously as possible.

Respectfully Submitted,

DITTHAVONG & CARLSON, P.C.

7/31/02  
Date



Phouphanomketh Ditthavong  
Attorney/Agent for Applicant(s)  
Reg. No. 44658

10507 Braddock Road  
Suite A  
Fairfax, VA 22032  
Tel. (703) 425-8508  
Fax. (703) 425-8518